

Base De Datos

Trabajo Práctico 1

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Base de Datos del Torneo Sudamericano de Basket

Integrante	LU	Correo electrónico
Aronson, Alex	443/08	alexaronson@gmail.com
Nahabedian, Leandro	250/08	leanahabedian@hotmail.com
Ravasi, Nicol	53/08	nravasi@gmail.com
Ocampo, Federico	599/02	faocampo2004@yahoo.com.ar

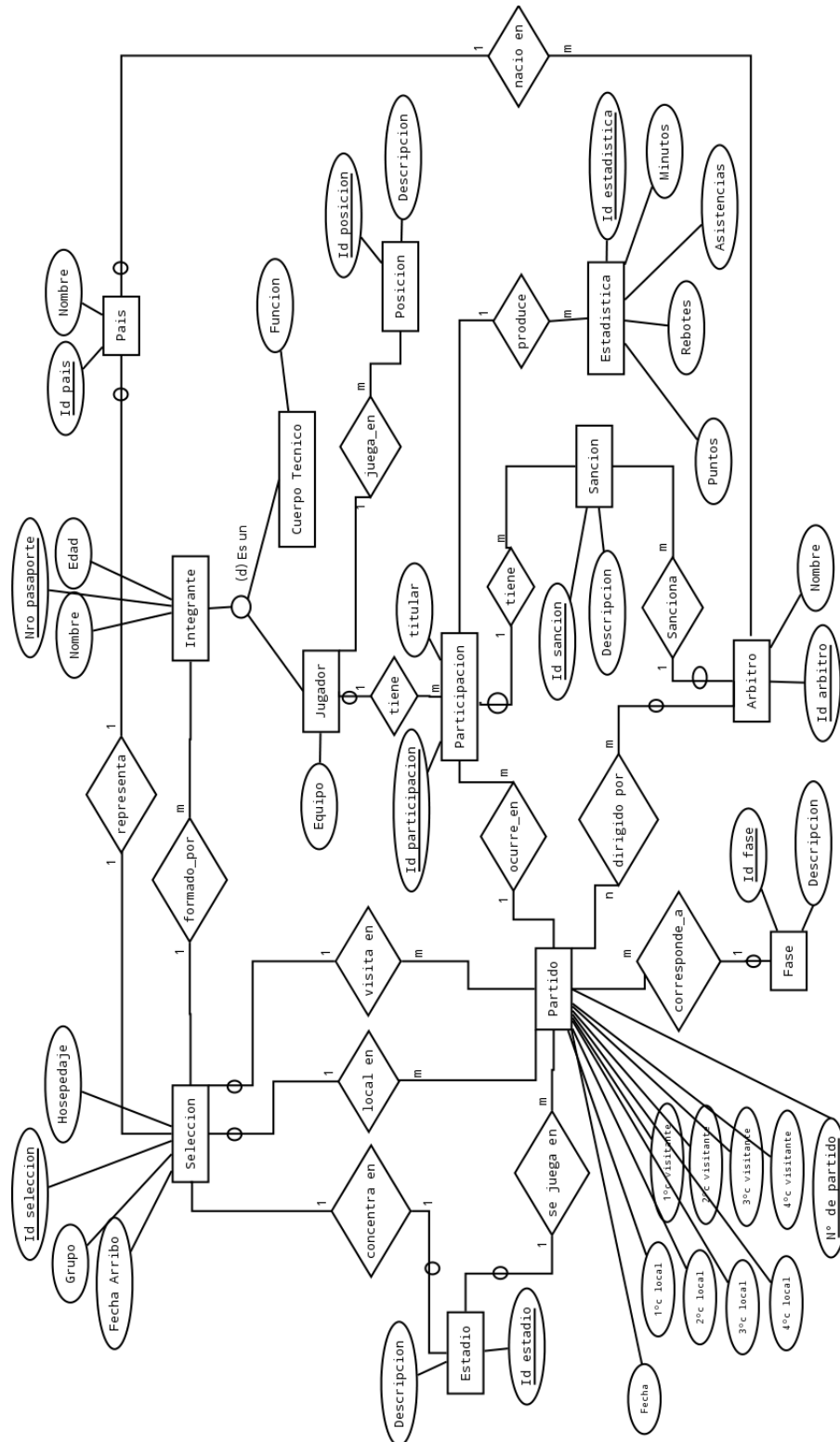
1. Introducción

El proceso de diseño de las relaciones de una base de datos involucra muchos conceptos tanto teóricos como prácticos relacionados con los diferentes aspectos involucrados; no solamente es necesario definir que entidades entrarán en juego y las interrelaciones entre si, sino también tener en cuenta aspectos de redundancia de datos y de definición de los esquemas de forma tal que se aprovechen las interrelaciones existentes.

La creación de esquemas de bases de datos a nivel industrial solamente exige ciertos niveles de formalidad y de correctitud, pero en el aspecto teórico existen muchos estudios que definen mejoras tanto para la reducción de redundancia de datos y dependencias como para la verificación de correctitud en la definición de los esquemas y las interrelaciones.

2. Diagrama Entidad-Relación (DER)

2.1. Diagrama



2.2. Decisiones de Diseño

Para comenzar con el trabajo lo primero que hicimos fue tratar de identificar en el texto todo aquello que considerabamos que podía llegar a ser una entidad, para ellos seleccionamos del texto las palabras claves tales como, Jugador, Estadística, Partido, Selección etc. Luego detectamos que había atributos tales, como país y estadio que se usaban de manera repetidas, por lo que decidimos pasar a estas también en forma de Entidades. Una vez con las entidades definidas nos propusimos llevar adelante la correcta diagramación de las interrelaciones. Si bien no encontramos dificultades para poder relacionar las entidades, nos encontramos con una traba para poder relacionar al partido, con las estadísticas y con los jugadores. A primera instancia tratamos de trabajar con las estadísticas de forma que se generaban de manera agregada de la relación entre partido y jugador, sin embargo desistimos de esta opción ya que no es posible que no exista estadística alguna de un jugador que jugó un partido, por lo que no es necesario este tipo de relación. Intentamos hacer una relación de grado 3 sin embargo tampoco creímos que era la manera correcta porque nos obligaba a inicializar una planilla de estadística sobre todos los jugadores con valores en 0 y esto no es correcto, sino que las estadísticas se realizan sobre los jugadores que juegan. Finalmente decidimos implementar una entidad participación, encargada de administrar las estadísticas, si un jugador fue titular o no en ese partido y si tuvo o no sanciones. A medida que íbamos diagramando nos encontramos con la necesidad de tomar decisiones para que el diagrama creado sea leído de manera correcta.

- Decidimos implementar la relación de partido con selección con dos relaciones distintas, la cual uno pertenece a una selección que hace el modo de Local y la otra de Visitante, es importante indicar que se debe restringir que esta selección no sea la misma, ya que esto es imposible en la realidad.
- Por partido solo puede haber una cantidad de 12 jugadores correspondientes a cada equipo, es decir que si bien la relación es en base a m este se limita a solamente 12
- Se debe restringir que la cantidad de titulares de un partido sea igual a 5 tal como lo indica el trabajo para ello creamos un trigger en el cual si se intenta cargar un jugador como titular y ya se alcanzó la cantidad máxima, esa inserción falla.
- Asumimos que la posición de un jugador es propia del mismo e independientemente del partido, tal como es en la realidad de las estadísticas del basquet.
- Permitimos que haya más de un árbitro por partido, ya que entendemos que el enunciado así lo pide refiriéndose constantemente al término arbitraje como plural
- Si bien el partido tiene muchos árbitros, una sanción puede ser aplicada por un solo árbitro.
- Decidimos no colocar un campo de resultado final dentro de la tabla partido ya que este campo se puede deducir de la suma de los cuartos correspondientes a cada selección.
- Si bien el trabajo solicita que se almacenen solo las estadísticas de los mejores jugadores, también pide como ítem emitir un reporte de las mismas sobre todos los que jugaron, por lo que no solo almacenamos información de los mejores sino que la haremos sobre todos los jugadores con partidos en su haber.

3. Modelo Relacional

3.1. Esquemas de Relaciones

En las siguientes secciones se formalizan las entidades y relaciones propuestas en el **Diagrama Entidad-Relación** mediante el esquema de **Modelo Relacional**, definiendo claves candidatas, primarias y foráneas para cada una de ellas. Por otro lado definiremos las restricciones que tiene nuestro esquema.

Seleccion(IdSeleccion, Grupo, Hospedaje, Fecha Arribo, IdPais, IdEstadio)
 PK = CK = IdSeleccion
 FK = IdPais, IdEstadio

Pais(IdPais, Nombre)
 PK = IdPais
 CK = IdPais, Nombre

Integrante(NroPasaporte, Nombre, Edad, EsUn, IdSeleccion)
 PK = CK = NroPasaporte
 FK = IdSeleccion

Cuerpo Tecnico (NroPasaporte, Funcion)
 PK = CK = FK = NroPasaporte

Jugador(NroPasaporte, Equipo, IdPosicion)
 PK = CK = NroPasaporte
 FK = NroPasaporte, IdPosicion

Participacion(IdParticipacion, EsTitular, NroPasaporte, NroDePartido)
 PK=CK = IdParticipacion
 FK= NroPasaporte, NroDePartido

Sancion(IdSancion, Descripcion, IdParticipacion ,IdArbitro)
 PK=CK = IdSancion
 FK= IdParticipacion, IdArbitro

Estadistica(IdEstadistica, Minutos, Asistencias, Rebotes, Puntos, IdParticipacion)
 PK=CK = IdEstadistica
 FK=IdParticipacion

Arbitro(IdArbitro, Nombre, IdPais)

PK=CK=IdArbitro

FK=IdPais

Partido(NroPartido, 1CuartoLocal, 2CuartoLocal, 3CuartoLocal, 4CuartoLocal, 1CuartoVisitante, 2CuartoVisitante, 3CuartoVisitante, 4CuartoVisitante, Fecha, IdFase, IdArbitro, IdSeleccionLocal, IdSeleccionVisitante, IdEstadio)

PK=CK= NroPartido

FK=IdFase, IdArbitro, IdSeleccionLocal, IdSeleccionVisitante, IdEstadio

Fase(IdFase, Descripcion)

PK=CK=IdFase

Estadio(IdEstadio, Descripcion)

PK=CK=IdEstadio

Posicion(IdPosicion, Descripcion)

PK=CK=IdPosicion

DirigidoPor(nroPartido, idArbitro)

PK = CK = (nroPartido, idArbitro)

FK = nroPartido, idArbitro

4. Funcionalidades de la Base de Datos

4.1. Introducción

En este trabajo tuvimos que realizar stored procedures y triggers para poder indicar los países que iutulizaron a todos sus jugadores como local, luego tuvimos que efectuar un listado de estadísticas indicando el promedio de las mismas para todos los jugadores que tuvieron participación, y finalmente nos encontramos con la necesidad de entregar una lista de árbitros permitidos para dirigir un partido creado. Por otro lado insertamos restricciones necesarias para que un arbitro no pueda dirigir partidos que juega la selección de su país, y también para evitar una incorrecta carga de datos en el sistema. Esto lo hicimos a través de triggers que fallan si no cumplen con las condiciones

4.2. Stored Procedures

4.2.1. Procedure 1

```
USE 'TP1BD';
DROP procedure IF EXISTS 'Selecciones_Usaron_Todos_Jugs';

DELIMITER $$
USE 'TP1BD'$$
CREATE PROCEDURE 'TP1BD'.'Selecciones_Usaron_Todos_Jugs' ()
BEGIN

select
    p.nombre
from
    Seleccion s,
    Pais p
where
    s.idPais = p.idPais
    and not exists( select
        i.idSeleccion
    from
        Jugador j, Integrante i
    where
        s.idSeleccion = i.idSeleccion
        and j.nroPasaporte = i.nroPasaporte
        and j.nroPasaporte not in (select distinct
            p.nroPasaporte
        from
            participacion p
```

```

        where
            p.esTitular))
and exists (select * from Partido p  where p.idSeleccionLocal = s.idSeleccion or p.idSeleccionVisi

END$$

DELIMITER ;

```

4.2.2. Estadística Jugadores

```

USE 'tp1bd';
DROP procedure IF EXISTS 'Estadisticas_Jugadores';

DELIMITER $$
USE 'tp1bd'$$
CREATE PROCEDURE 'tp1bd'.'Estadisticas_Jugadores' ()
BEGIN
select
    Nombre,
    Partidos_Jugados,
    Minutos,
    Puntos,
    Rebotes,
    Asistencias,
    (Puntos / Partidos_Jugados) Ptos_por_partido,
    (Asistencias / Partidos_Jugados) Asistencias_por_partido,
    (Rebotes / Partidos_Jugados) Rebotes_por_partido
from
    (select
        p.nroPasaporte,
        i.Nombre,
        sum(minutos) Minutos,
        sum(rebotes) Rebotes,
        sum(puntos) Puntos,
        sum(asistencias) Asistencias,
        count(1) Partidos_Jugados
    from
        integrante i, estadistica e, participacion p
    where
        p.idParticipacion = e.idParticipacion
        and i.nroPasaporte = p.nroPasaporte
    group by p.nroPasaporte , i.Nombre

```



```
        order by Partidos_Jugados) Total
order by partidos_jugados desc;
END$$
```

```
DELIMITER ;
```

4.2.3. Trigger Valida Arbitro

```
-- Define un nuevo caracter delimitador en lugar del ','
DELIMITER $$
CREATE
    TRIGGER 'validarAsignacionArbitro' BEFORE INSERT
    ON 'tp1bd'.'dirigidopor'

FOR EACH ROW BEGIN
    DECLARE idPaisArbitro INT(11);
        DECLARE idPaisLocal INT(11);
    DECLARE idPaisVisita INT(11);

    -- Id del Pais del Arbitro
    SELECT a.idPais INTO @idPaisArbitro
    FROM arbitro a
    WHERE a.idArbitro = NEW.idArbitro;

    -- Ids de los paises de los equipos local y visit
    SELECT s1.idPais, s2.idPais
    INTO @idPaisLocal, @idPaisVisita
    FROM seleccion s1, seleccion s2, partido p
    WHERE
        p.nroPartido = NEW.nroPartido
        AND s1.idPais = p.idSeleccionLocal
        AND s2.idPais = p.idSeleccionVisitante;

        IF @idPaisArbitro = @idPaisLocal OR @idPaisArbitro = @idPaisVisita THEN
            -- Para invalidar la operacion, hacemos un INSERT de NULL invalido
            SET NEW.nroPartido = NULL;
        END IF;
    END;$$
DELIMITER ;

-- Se repone el caracter delimitar por defecto
```

4.2.4. Trigger Valida Paises Diferentes

```
-- Define un nuevo caracter delimitador en lugar del ';'
DELIMITER $$
CREATE
    TRIGGER 'validarPaisesDiferentes' BEFORE INSERT
    ON 'tp1bd'.'Partido'

FOR EACH ROW BEGIN

    IF NEW.idSeleccionLocal = NEW.idSeleccionVisitante THEN
        -- Para invalidar la operacion, hacemos un INSERT de NULL invalido
        SET NEW.nroPartido = NULL;
    END IF;
END;$$
DELIMITER ;
-- Se repone el caracter delimitar por defecto
```

4.2.5. Trigger valida Participaciones

```
-- Define un nuevo caracter delimitador en lugar del ';'
DELIMITER $$
CREATE
    TRIGGER 'validarParticipaciones' BEFORE INSERT
    ON 'tp1bd'.'participacion'

FOR EACH ROW BEGIN
DECLARE cantTitulares INT(11);
DECLARE cantJugadores INT(11);
DECLARE idPaisJugador INT(11);

SELECT j.idSeleccion INTO @idPaisJugador
        FROM integrante j
        WHERE j.nroPasaporte = NEW.nroPasaporte;

SELECT count(1) Cantidad_Jugadores
INTO @cantJugadores
FROM participacion p , integrante i
WHERE p.nroPasaporte = i.nroPasaporte and p.nroPartido = NEW.nroPartido and i.idSeleccion = @idPaisJugador
GROUP BY idSeleccion, nroPartido;

SELECT count(1) Cantidad_Jugadores
INTO @cantTitulares
```

```
FROM participacion p , integrante i
WHERE p.nroPasaporte = i.nroPasaporte and p.nroPartido = NEW.nroPartido and i.idSeleccion = @idPa
and esTitular
GROUP BY idSeleccion, nroPartido;

    IF @cantJugadores >= 12 or (@cantTitulares >=5 and new.esTitular) THEN
        -- Para invalidar la operacion, hacemos un INSERT de NULL invalido
        SET NEW.idParticipacion = NULL;
    END IF;
END;$$
DELIMITER ;
-- Se repone el caracter delimitar por defecto
```

Referencias

- [RG97] Raghu Ramakrishnan and Johannes Gehrke. Database Management Systems. McGrawHill, second edition edition, 1997.
- [DB07] Cátedra Bases de Datos. Apunte de modelización. Apunte de Modelización, Marzo 2009.
- [TYF86] Toby J. Teory, Dongqing Yang, and James P. Fry. A logical design methodology for relational databases using the extended entity-relationship model. Computing Surveys, 18(2), jun 1986.
- [MY09] MySQL Official Documentation. <http://dev.mysql.com>.